Analyze_ab_test_results_notebook

September 6, 2020

0.1 Analyze A/B Test Results

You may either submit your notebook through the workspace here, or you may work from your local machine and submit through the next page. Either way assure that your code passes the project RUBRIC. **Please save regularly.**

This project will assure you have mastered the subjects covered in the statistics lessons. The hope is to have this project be as comprehensive of these topics as possible. Good luck!

0.2 Table of Contents

Introduction

A/B tests are very commonly performed by data analysts and data scientists. It is important that you get some practice working with the difficulties of these

For this project, you will be working to understand the results of an A/B test run by an e-commerce website. Your goal is to work through this notebook to help the company understand if they should implement the new page, keep the old page, or perhaps run the experiment longer to make their decision.

As you work through this notebook, follow along in the classroom and answer the corresponding quiz questions associated with each question. The labels for each classroom concept are provided for each question. This will assure you are on the right track as you work through the project, and you can feel more confident in your final submission meeting the criteria. As a final check, assure you meet all the criteria on the RUBRIC.

Part I - Probability

To get started, let's import our libraries.

```
In [1]: import pandas as pd
        import numpy as np
        import random
        import matplotlib.pyplot as plt
        %matplotlib inline
        #We are setting the seed to assure you get the same answers on quizzes as we set up
        random.seed(42)
```

1. Now, read in the ab_data.csv data. Store it in df. **Use your dataframe to answer the questions in Quiz 1 of the classroom.**

a. Read in the dataset and take a look at the top few rows here:

```
In [2]: df = pd.read_csv('ab_data.csv')
        df.head(4)
```

Out[2]:

	user_id	timestamp	group	landing_page	converted
0	851104	2017-01-21 22:11:48.556739	control	old_page	0
1	804228	2017-01-12 08:01:45.159739	control	old_page	0
2	661590	2017-01-11 16:55:06.154213	treatment	new_page	0
3	853541	2017-01-08 18:28:03.143765	treatment	new_page	0

b. Use the cell below to find the number of rows in the dataset.

```
In [3]: rows=df.shape[0]
        print('Number of rows in dataset :: ',rows)
        #find no of columns and non-empty fields in dataset
        print('dataset information :: ')
        df.info()
        df.head(10)
```

```
Number of rows in dataset ::  294478
dataset information ::
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 294478 entries, 0 to 294477
Data columns (total 5 columns):
user_id         294478 non-null int64
timestamp       294478 non-null object
group           294478 non-null object
landing_page    294478 non-null object
converted       294478 non-null int64
dtypes: int64(2), object(3)
memory usage: 11.2+ MB
```

Out[3]:

	user_id	timestamp	group	landing_page	converted
0	851104	2017-01-21 22:11:48.556739	control	old_page	0
1	804228	2017-01-12 08:01:45.159739	control	old_page	0
2	661590	2017-01-11 16:55:06.154213	treatment	new_page	0
3	853541	2017-01-08 18:28:03.143765	treatment	new_page	0
4	864975	2017-01-21 01:52:26.210827	control	old_page	1
5	936923	2017-01-10 15:20:49.083499	control	old_page	0
6	679687	2017-01-19 03:26:46.940749	treatment	new_page	1
7	719014	2017-01-17 01:48:29.539573	control	old_page	0
8	817355	2017-01-04 17:58:08.979471	treatment	new_page	1
9	839785	2017-01-15 18:11:06.610965	treatment	new_page	1

c. The number of unique users in the dataset.

```
In [4]: print('Number of unique users in the dataset :: ',df['user_id'].nunique())
```

```
Number of unique users in the dataset ::  290584
```

d. The proportion of users converted.

```
In [5]: print('Proportion of users converted :: ',df['converted'].mean())
```

```
Proportion of users converted ::  0.119659193556
```

e. The number of times the `new_page` and `treatment` don't match.

```
In [6]: print('Number of times new_page and treatment dont line up :: ',df.query('landing_page =
```

```
Number of times new_page and treatment dont line up ::  3893
```

f. Do any of the rows have missing values?

```
In [7]: #counts null values in all columns
        df.isnull().sum()

Out[7]: user_id         0
        timestamp       0
        group           0
        landing_page    0
        converted       0
        dtype: int64
```

2. For the rows where **treatment** does not match with **new_page** or **control** does not match with **old_page**, we cannot be sure if this row truly received the new or old page. Use **Quiz 2** in the classroom to figure out how we should handle these rows.

a. Now use the answer to the quiz to create a new dataset that meets the specifications from the quiz. Store your new dataframe in **df2**.

```
In [8]: df2=df
        df2.drop(df.query("(group == 'treatment' and landing_page == 'old_page') or (group == 'c
```

```
In [9]: # Double Check all of the correct rows were removed - this should be 0
        df2[((df2['group'] == 'treatment') == (df2['landing_page'] == 'new_page')) == False].sha

Out[9]: 0
```

3. Use **df2** and the cells below to answer questions for **Quiz3** in the classroom.

a. How many unique **user_id**s are in **df2**?

```
In [10]: print('Number of unique users in the dataset :: ',df2['user_id'].nunique())
```

```
Number of unique users in the dataset ::  290584
```

b. There is one **user_id** repeated in **df2**. What is it?

```
In [11]: df2[df2.duplicated('user_id')]
```

Out[11]:

	user_id	timestamp	group	landing_page	converted
2893	773192	2017-01-14 02:55:59.590927	treatment	new_page	0

c. What is the row information for the repeat **user_id**?

```
In [12]: df2[df2.user_id.duplicated(keep=False)]
```

Out[12]:

	user_id	timestamp	group	landing_page	converted
1899	773192	2017-01-09 05:37:58.781806	treatment	new_page	0
2893	773192	2017-01-14 02:55:59.590927	treatment	new_page	0

d. Remove **one** of the rows with a duplicate **user_id**, but keep your dataframe as **df2**.

```
In [13]: df2.drop_duplicates('user_id', inplace=True)
         df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 290584 entries, 0 to 294477
Data columns (total 5 columns):
user_id         290584 non-null int64
timestamp       290584 non-null object
group           290584 non-null object
landing_page    290584 non-null object
converted       290584 non-null int64
dtypes: int64(2), object(3)
memory usage: 13.3+ MB
```

4. Use **df2** in the cells below to answer the quiz questions related to **Quiz 4** in the classroom.

a. What is the probability of an individual converting regardless of the page they receive?

```
In [14]: df2['converted'].mean()
```

```
Out[14]: 0.11959708724499628
```

b. Given that an individual was in the `control` group, what is the probability they converted?

```
In [15]: df2[df2['group'] == 'control']['converted'].mean()
```

```
Out[15]: 0.1203863045004612
```

c. Given that an individual was in the `treatment` group, what is the probability they converted?

```
In [16]: df2[df2['group'] == 'treatment']['converted'].mean()
```

```
Out[16]: 0.11880806551510564
```

d. What is the probability that an individual received the new page?

```
In [17]: print('New page probability :',(df2.landing_page == "new_page").mean())
```

```
New page probability : 0.500061944223
```

```
In [18]: print('Old page probability :',(df2.landing_page == "old_page").mean())
```

```
Old page probability : 0.499938055777
```

e. Consider your results from parts (a) through (d) above, and explain below whether you think there is sufficient evidence to conclude that the new treatment page leads to more conversions.

From above results we can find below information :

a) probability of conversion : 0.11959708724499628

b) probability of conversion when individual was in the control group : 0.1203863045004612

c) probability of conversion when individual was in the treatment group : 0.11880806551510564

d) probability of individual receiving a new page : 0.500061944223

We cannot declare that the new treatment page leads to more conversions due to no sufficient evidence as the variation between the groups of treatment and control is minute and almost equal from the results obtained. Also probability of new page is roughly 50% which doesn't prove that all new treatment page will be more converted.

Part II - A/B Test

Notice that because of the time stamp associated with each event, you could technically run a hypothesis test continuously as each observation was observed.

However, then the hard question is do you stop as soon as one page is considered significantly better than another or does it need to happen consistently for a certain amount of time? How long do you run to render a decision that neither page is better than another?

These questions are the difficult parts associated with A/B tests in general.

`1.` For now, consider you need to make the decision just based on all the data provided. If you want to assume that the old page is better unless the new page proves to be definitely better at a Type I error rate of 5%, what should your null and alternative hypotheses be? You can state your hypothesis in terms of words or in terms of p_{old} and p_{new}, which are the converted rates for the old and new pages.

H0: =

H1: !=

`2.` Assume under the null hypothesis, p_{new} and p_{old} both have "true" success rates equal to the **converted** success rate regardless of page - that is p_{new} and p_{old} are equal. Furthermore, assume they are equal to the **converted** rate in **ab_data.csv** regardless of the page.

Use a sample size for each page equal to the ones in **ab_data.csv**.

Perform the sampling distribution for the difference in **converted** between the two pages over 10,000 iterations of calculating an estimate from the null.

Use the cells below to provide the necessary parts of this simulation. If this doesn't make complete sense right now, don't worry - you are going to work through the problems below to complete this problem. You can use **Quiz 5** in the classroom to make sure you are on the right track.

a. What is the **conversion rate** for p_{new} under the null?

```
In [19]: p_new = df2['converted'].mean()
         print('Convert rate for p_new under the null : ',p_new)
```

```
Convert rate for p_new under the null :  0.119597087245
```

b. What is the **conversion rate** for p_{old} under the null?

```
In [20]: p_old = df2['converted'].mean()
         print('Convert rate for p_old under the null : ',p_old)
```

```
Convert rate for p_old under the null :  0.119597087245
```

c. What is n_{new}, the number of individuals in the treatment group?

```
In [21]: n_new = df2.query("group == 'treatment'").shape[0]
         print('n_new :: ', n_new)
```

```
n_new ::  145310
```

d. What is n_{old}, the number of individuals in the control group?

```
In [22]: n_old = df2.query("group == 'control'").shape[0]
         print('n_old :: ', n_old)
```

```
n_old ::  145274
```

e. Simulate n_{new} transactions with a conversion rate of p_{new} under the null. Store these n_{new} 1's and 0's in **new_page_converted**.

```
In [23]: new_page_converted = np.random.binomial(n_new,p_new)
```

f. Simulate n_{old} transactions with a conversion rate of p_{old} under the null. Store these n_{old} 1's and 0's in **old_page_converted**.

```
In [24]: old_page_converted = np.random.binomial(n_old,p_old)
```

g. Find p_{new} - p_{old} for your simulated values from part (e) and (f).

```
In [25]: new_page_converted/n_new - old_page_converted/n_old
```

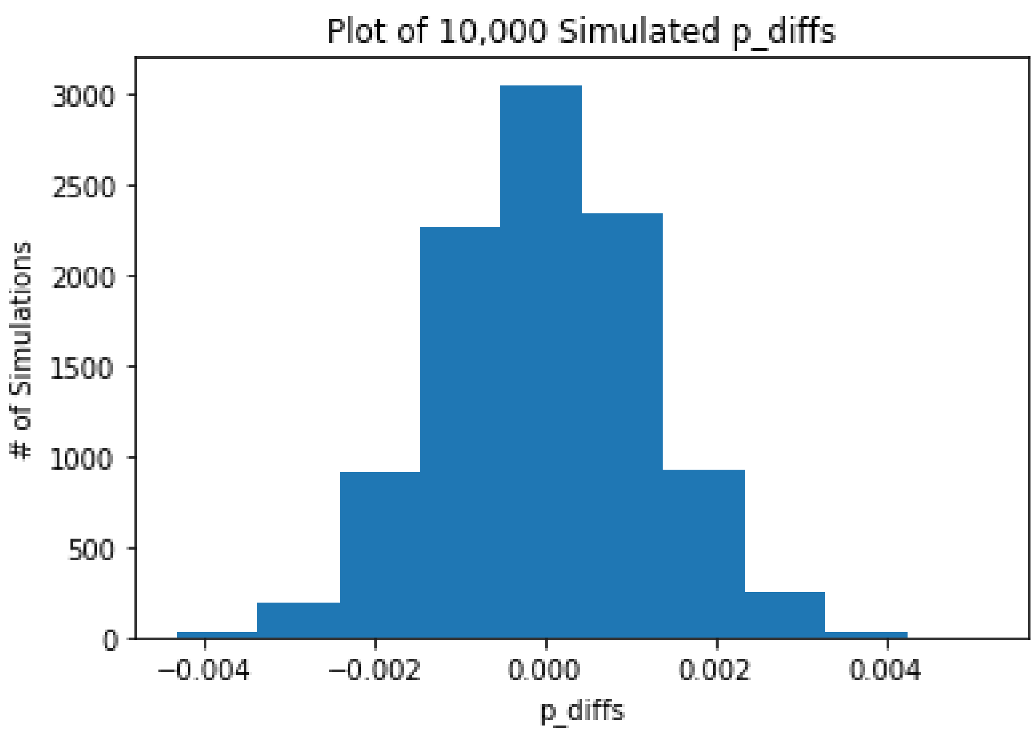
```
Out[25]: 0.0010372298347344766
```

h. Create 10,000 p_{new} - p_{old} values using the same simulation process you used in parts (a) through (g) above. Store all 10,000 values in a NumPy array called **p_diffs**.

```
In [26]: p_diffs = []
         for _ in range(10000):
             new_page_converted = np.random.binomial(n_new,p_new)
             old_page_converted = np.random.binomial(n_old, p_old)
             p_diff = new_page_converted/n_new - old_page_converted/n_old
             p_diffs.append(p_diff)
```

i. Plot a histogram of the **p_diffs**. Does this plot look like what you expected? Use the matching problem in the classroom to assure you fully understand what was computed here.

```
In [27]: plt.hist(p_diffs);
         plt.ylabel('# of Simulations')
         plt.xlabel('p_diffs')
         plt.title('Plot of 10,000 Simulated p_diffs');
```



j. What proportion of the **p_diffs** are greater than the actual difference observed in **ab_data.csv**?

```
In [28]: p_diff_orig = df[df['landing_page'] == 'new_page']['converted'].mean() -  df[df['landin
         print('p_diff from ab_data.csv : ',p_diff_orig)
```

```
p_diff from ab_data.csv :  -0.00157823898536
```

```
In [29]: p_diffs = np.array(p_diffs)
         p_diff_proportion = (p_diff_orig < p_diffs).mean()
         print('proportion of p_diffs greater than p_diffs from ab_data.csv : ',p_diff_proportio
```

```
proportion of p_diffs greater than p_diffs from ab_data.csv :  0.9062
```

k. Please explain using the vocabulary you've learned in this course what you just computed in part **j.** What is this value called in scientific studies? What does this value mean in terms of whether or not there is a difference between the new and old pages?

We are calculating pvalue. If null hypothesis H0 is true pvalue gives the probability of statistics tested. In this case, the given computed p-value is too high such that it suggests that we fail to reject the null-hypothesis. So,the new page doesn't have better conversion rates than the old page.

l. We could also use a built-in to achieve similar results. Though using the built-in might be easier to code, the above portions are a walkthrough of the ideas that are critical to correctly thinking about statistical significance. Fill in the below to calculate the number of conversions for each page, as well as the number of individuals who received each page. Let `n_old` and `n_new` refer the the number of rows associated with the old page and new pages, respectively.

```
In [30]: import statsmodels.api as sm

         convert_old = df2.query('group == "control"').converted.sum()
         convert_new = df2.query('group == "treatment"').converted.sum()
         n_old = df2.query("landing_page == 'old_page'").shape[0]
         n_new = df2.query("landing_page == 'new_page'").shape[0]
```

```
/opt/conda/lib/python3.6/site-packages/statsmodels/compat/pandas.py:56: FutureWarning: The panda
  from pandas.core import datetools
```

m. Now use `stats.proportions_ztest` to compute your test statistic and p-value. Here is a helpful link on using the built in.

```
In [31]: z_score, p_value = sm.stats.proportions_ztest(count = [convert_new,convert_old], nobs =
         print ("z_score:",z_score)
         print("p_value:",p_value)
```

```
z_score: -1.31092419842
p_value: 0.905058312759
```

```
In [32]: from scipy.stats import norm
         # significant of z-score
         print(norm.cdf(z_score))

         # for our single-sides test, assumed at 95% confidence level, we calculate:
         print(norm.ppf(1-(0.05)))
```

```
0.094941687241
1.64485362695
```

n. What do the z-score and p-value you computed in the previous question mean for the conversion rates of the old and new pages? Do they agree with the findings in parts **j.** and **k.**?

zscore is a measure of how many standard deviations below or above the population mean a raw score is. We find that the z-score of 1.3109241984234394 is less than the critical value of 1.6448536269514722 which means we can't reject the null hypothesis. we find that old page conversions are slightly better than new page conversions. Eventhough the values are different from findings in parts j and k but it suggests there is no significant difference between old page and new page conversions. so there is no evidence gathered to reject the null hypothesis but does include in highest rate of probability of being null hypothesis. ### Part III - A regression approach

`1.` In this final part, you will see that the result you achieved in the A/B test in Part II above can also be achieved by performing regression.

a. Since each row is either a conversion or no conversion, what type of regression should you be performing in this case?

Logistic Regression

b. The goal is to use **statsmodels** to fit the regression model you specified in part **a.** to see if there is a significant difference in conversion based on which page a customer receives. However, you first need to create in df2 a column for the intercept, and create a dummy variable column for which page each user received. Add an **intercept** column, as well as an **ab_page** column, which is 1 when an individual receives the **treatment** and 0 if **control**.

```
In [33]: df2['intercept'] = 1
         df2[['ab_page', 'old_page']] = pd.get_dummies(df2['landing_page'])
         df2.head()
```

```
Out[33]:    user_id                   timestamp      group landing_page  converted  \
         0   851104  2017-01-21 22:11:48.556739    control     old_page          0
         1   804228  2017-01-12 08:01:45.159739    control     old_page          0
         2   661590  2017-01-11 16:55:06.154213  treatment     new_page          0
         3   853541  2017-01-08 18:28:03.143765  treatment     new_page          0
         4   864975  2017-01-21 01:52:26.210827    control     old_page          1
```

```
   intercept  ab_page  old_page
0          1        0         1
1          1        0         1
2          1        1         0
3          1        1         0
4          1        0         1
```

c. Use **statsmodels** to instantiate your regression model on the two columns you created in part b., then fit the model using the two columns you created in part **b.** to predict whether or not an individual converts.

```
In [38]: import statsmodels.api as sm
         import scipy.stats as stats
         logit = sm.Logit(df2['converted'],df2[['intercept' ,'ab_page']])
         results = logit.fit()
```

```
Optimization terminated successfully.
         Current function value: 0.366118
         Iterations 6
```

d. Provide the summary of your model below, and use it as necessary to answer the following questions.

```
In [39]: stats.chisqprob = lambda chisq, df: stats.chi2.sf(chisq, df)
         results.summary()
```

```
Out[39]: <class 'statsmodels.iolib.summary.Summary'>
         """
                                    Logit Regression Results
         ==============================================================================
         Dep. Variable:              converted   No. Observations:               290584
         Model:                          Logit   Df Residuals:                   290582
         Method:                           MLE   Df Model:                            1
         Date:                Sun, 06 Sep 2020   Pseudo R-squ.:               8.077e-06
         Time:                        17:14:31   Log-Likelihood:            -1.0639e+05
         converged:                       True   LL-Null:                   -1.0639e+05
                                                 LLR p-value:                   0.1899
         ==============================================================================
                          coef    std err          z      P>|z|      [0.025      0.975]
         ------------------------------------------------------------------------------
         intercept     -1.9888      0.008   -246.669      0.000      -2.005      -1.973
         ab_page       -0.0150      0.011     -1.311      0.190      -0.037       0.007
         ==============================================================================
         """
```

e. What is the p-value associated with **ab_page**? Why does it differ from the value you found in **Part II**? **Hint**: What are the null and alternative hypotheses associated with your regression model, and how do they compare to the null and alternative hypotheses in **Part II**?

The p-value here suggests that that new page is not statistically significant as 0.19 > 0.05 . In this section it was a two sided test and in Part II it was a one sided test. The p-value is much greater than the Part II so we cannot reject the null hypothesis. So, we cannot reject the null hypothesis.

f. Now, you are considering other things that might influence whether or not an individual converts. Discuss why it is a good idea to consider other factors to add into your regression model. Are there any disadvantages to adding additional terms into your regression model?

There might be many factors that can effect individual converts; such as demographic factors can play a significant change. Also usage and access may effect the rate of conversion. We can find new trends using other factors but there may be some disadvantages like even with new factors we may miss some other influencing factors which lead to unreliable and contradictory results compared to previous results.

g. Now along with testing if the conversion rate changes for different pages, also add an effect based on which country a user lives in. You will need to read in the **countries.csv** dataset and merge together your datasets on the appropriate rows. Here are the docs for joining tables.

Does it appear that country had an impact on conversion? Don't forget to create dummy variables for these country columns - **Hint: You will need two columns for the three dummy variables.** Provide the statistical output as well as a written response to answer this question.

In [40]:
```
countries_df = pd.read_csv('./countries.csv')
df_new = countries_df.set_index('user_id').join(df2.set_index('user_id'), how='inner')
```

In [41]:
```
### Create the necessary dummy variables
df_new[['CA', 'US']] = pd.get_dummies(df_new['country'])[['CA','US']]
df_new.head()
```

Out[41]:
```
        country                   timestamp      group landing_page  \
user_id
834778       UK  2017-01-14 23:08:43.304998    control     old_page
928468       US  2017-01-23 14:44:16.387854  treatment     new_page
822059       UK  2017-01-16 14:04:14.719771  treatment     new_page
711597       UK  2017-01-22 03:14:24.763511    control     old_page
710616       UK  2017-01-16 13:14:44.000513  treatment     new_page

         converted  intercept  ab_page  old_page  CA  US
user_id
834778           0          1        0         1   0   0
928468           0          1        1         0   0   1
822059           1          1        1         0   0   0
711597           0          1        0         1   0   0
710616           0          1        1         0   0   0
```

In [42]:
```
### Fit Your Linear Model And Obtain the Results
df_new['intercept'] = 1
```

```
log_mod = sm.Logit(df_new['converted'], df_new[['intercept', 'CA', 'US', 'ab_page']])
results = log_mod.fit()
results.summary()
```

```
Optimization terminated successfully.
         Current function value: 0.366113
         Iterations 6
```

Out[42]:

```
<class 'statsmodels.iolib.summary.Summary'>
"""
                           Logit Regression Results
==============================================================================
Dep. Variable:              converted   No. Observations:               290584
Model:                          Logit   Df Residuals:                   290580
Method:                           MLE   Df Model:                            3
Date:                Sun, 06 Sep 2020   Pseudo R-squ.:               2.323e-05
Time:                        17:14:48   Log-Likelihood:            -1.0639e+05
converged:                       True   LL-Null:                   -1.0639e+05
                                        LLR p-value:                    0.1760
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
intercept     -1.9794      0.013   -155.415      0.000      -2.004      -1.954
CA            -0.0506      0.028     -1.784      0.074      -0.106       0.005
US            -0.0099      0.013     -0.743      0.457      -0.036       0.016
ab_page       -0.0149      0.011     -1.307      0.191      -0.037       0.007
==============================================================================
"""
```

h. Though you have now looked at the individual factors of country and page on conversion, we would now like to look at an interaction between page and country to see if there significant effects on conversion. Create the necessary additional columns, and fit the new model.

Provide the summary results, and your conclusions based on the results.

In [43]:

```
df_new['US_ab_page'] = df_new['US'] * df_new['ab_page']
df_new['CA_ab_page'] = df_new['CA'] * df_new['ab_page']
df_new.head()
```

Out[43]:

```
        country                   timestamp      group landing_page  \
user_id
834778       UK  2017-01-14 23:08:43.304998    control     old_page
928468       US  2017-01-23 14:44:16.387854  treatment     new_page
822059       UK  2017-01-16 14:04:14.719771  treatment     new_page
711597       UK  2017-01-22 03:14:24.763511    control     old_page
710616       UK  2017-01-16 13:14:44.000513  treatment     new_page

         converted  intercept  ab_page  old_page  CA  US  US_ab_page  \
```

```
user_id
834778              0          1          0          1    0    0            0
928468              0          1          1          0    0    1            1
822059              1          1          1          0    0    0            0
711597              0          1          0          1    0    0            0
710616              0          1          1          0    0    0            0

         CA_ab_page
user_id
834778            0
928468            0
822059            0
711597            0
710616            0
```

0.3 Conclusions

On applying regression for the above values, it is observed that the p-value factor between US and Canada are varied highly, where it is high in US than Canada. THis is because of users likeliness to convert. But there is no complete evidence of the null hypothesis for it to be rejected. On observation of the total analysis performance,it can be stated that the new page does not show much variation in the histogram and the old page is much better than the new page in accordance with the null hypothesis. We can accept Null Hypothesis as there is no significant difference in conversion rates. We can reject alternate hypothesis. These results are based on given dataset. There may be limitations due to incorrect data or missing columns.

```
In [44]: from subprocess import call
         call(['python', '-m', 'nbconvert', 'Analyze_ab_test_results_notebook.ipynb'])

Out[44]: 0
```